STUDENT MENTAL HEALTH DETECTION USING MACHINE LEARNING

A DESIGN PROJECT REPORT

submitted by

SAHAANA S

SHARVINI S

SWETHA M

THENNARASI M

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

STUDENT MENTAL HEALTH DETECTION USING MACHINE LEARNING

A DESIGN PROJECT REPORT

submitted by

SAHAANA S (811722104127)

SHARVINI S (811722104145)

SWETHA M (811722104166)

THENNARASI M (811722104168)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"STUDENT MENTAL HEALTH DETECTION USING MACHINE LEARNING"** is Bonafide work of **SAHAANA S (811722104127), SHARVINI S (811722104145), SWETHA M (811722104166), THENNARASI M (811722104168)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. A Delphin Carolina Rani, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR
Department of CSE
K Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

SIGNATURE
Ms. P. Karthika, M.E.,
SUPERVISOR
ASSISTANT PROFESSOR
Department of CSE
K Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

Submitted for the viva-voice examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"STUDENT MENTAL HEALTH DETECTION USING MACHINE LEARNING"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of Bachelor of Engineering. This project report is submitted on the partial fulfillment of the requirement of the award of Degree of Bachelor of Engineering.

Signature

SAHAANA S

SHARVINI S

SWETHA M

THENNARASI M

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution **"K RAMAKRISHNAN COLLEGE OF TECHNOLOGY"**, for providing us with the opportunity to do this project.

We are glad to credit and praise our honorable and respected chairman sir **Dr. K RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering adequate duration to complete it.

We would like to thank **Dr. N VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project with full satisfaction.

We heartily thank **Dr. A DELPHIN CAROLINA RANI, M.E., Ph.D.,** Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her support to pursue this project.

We express our deep and sincere gratitude and thanks to our project guide **Ms. P. KARTHIKA, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course. We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

The early detection of mental health issues enables specialists to provide more effective treatment, significantly enhancing student quality of life. Mental health encompasses psychological, emotional, and social well-being, profoundly affecting how individuals think, feel, and behave. It is vital at every life stage, from childhood and adolescence through adulthood, influencing overall functioning and well-being. This project focused on two machine learning techniques decision tree classifiers and random forest algorithms and evaluated their accuracy in detecting mental health problems. By employing various accuracy metrics, we aimed to identify the most effective method for early detection. The prediction of mental health issues using machine learning (ML) involves applying advanced algorithms to analyze patterns and predict the likelihood of mental health conditions. This approach leverages large datasets to identify key indicators and correlations that may not be easily recognizable through traditional methods. These models split data into branches based on specific criteria, forming a tree-like structure. At each node, the model decides the best feature to split the data on, aiming to maximize the distinction between classes Decision trees are intuitive and simple to understand but can over fit on complex datasets, reducing accuracy. Random forests area ensemble method that builds multiple decision trees and aggregates their predictions. Random forests are well-suited for handling large, noisy data sets and capturing subtle patterns in the data. The ultimate goal was to support targeted interventions and improve mental health outcomes across diverse populations, addressing the global need for scalable and precise mental health solutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION		FULL FORM
PTSD	-	Post-Traumatic Stress Disorder
NLP	-	Natural Language Processing
SVM	-	Support Vector Machine
CNN	-	Convolutional Neural Network
RNN	-	Recurrent Neural Network
ML	-	Machine Learning
Colab	-	Co-laboratory
IDE	-	Interactive Development Environment
GPU	-	Graphics Processing Unit
TPU	-	Tensor Processing Unit
CSV	-	Comma-Seperated values
SQL	-	Structured Query Languagen
DFD	-	Data Flow Diagram
LSTM	-	Long Short-Term Memory

CHAPTER 1

INTRODUCTION

1.1 DETECTING MENTAL HEALTH OF AN STUDENT

Detecting mental health issues using machine learning involves a comprehensive and structured approach that begins with clearly defining the problem typically aimed at identifying individuals who may be experiencing conditions such as depression, anxiety, or post-traumatic stress disorder (PTSD). The first step is data collection, which is often drawn from various sources such as social media platforms (e.g., Twitter, Reddit), wearable devices (e.g., smartwatches), audio/video recordings, clinical assessments, or responses to mental health questionnaires.These data sources provide valuable insights into behavioural, physiological, and emotional patterns relevant to mental health.

Once the data is gathered, it undergoes preprocessing to ensure quality and consistency. For text data, this involves tasks such as tokenization, stop-word removal, and text normalization. For physiological signals, such as heart rate or sleep patterns from wearable devices, preprocessing may include signal smoothing, normalization, and filtering to remove noise. If images or audio data are involved, techniques like resizing, feature extraction, or conversion to spectrograms (in the case of audio) are applied. In many cases, missing or inconsistent data must also be handled through imputation or by filtering out incomplete records.

1.2 OVERVIEW

Mental health detection using machine learning represents a rapidly advancing and promising field within healthcare technology, aiming to utilize sophisticated data-driven approaches to identify early signs and symptoms of psychological disorders such as depression, anxiety, stress, and other related mental health conditions. This innovative area capitalizes on the vast amount of data generated by individuals in their daily lives, harnessing it to provide insights that were previously difficult or impossible to obtain through conventional clinical methods. The process typically begins with the careful collection of diverse data types from multiple sources. These sources may include social media platforms where users share their thoughts and feelings, wearable devices that continuously monitor physiological signals like heart rate, sleep quality, and physical activity, as well as audio and video recordings that capture speech patterns, facial expressions, and other behavioral cues. Additionally, structured clinical assessments and psychological questionnaires provide standardized and validated data points that enrich the dataset with clinically relevant information.

On data acquisition, the collected raw data undergoes extensive preprocessing to prepare it for analysis. This involves cleaning the data by removing irrelevant or erroneous information, handling missing values, and normalizing formats to ensure consistency across different datasets. Subsequently, feature extraction techniques are employed to convert raw data into meaningful variables that can effectively represent underlying mental health indicators. For example, text data may be analyzed using natural language processing (NLP) to detect sentiment, emotion, and key themes, while physiological data might be processed to identify stress-related biomarkers or irregular sleep patterns.

Behavioral features such as changes in communication frequency or social withdrawal can also be quantified. These extracted features form the input for various machine learning models designed to detect subtle patterns and anomalies associated with mental health disorders.

The core of this approach lies in the development and training of machine learning algorithms, which range from traditional models like Support Vector Machines (SVM), Decision Trees, and Random Forests to more advanced deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These models learn from labeled datasets to distinguish between healthy and at-risk individuals by recognizing complex interactions among features that may not be evident to human observers.

After rigorous training and validation phases to optimize model accuracy, robustness, and generalizability, these systems can be integrated into practical applications, including mobile health applications, telemedicine platforms, and clinical decision support tools. Such implementations offer significant benefits by enabling continuous, non-invasive monitoring and early detection of mental health issues, thereby facilitating timely intervention and personalized care. Importantly, the field also addresses critical ethical challenges related to data privacy, informed consent, and the interpretability of AI decisions. Ensuring transparency and safeguarding sensitive personal information are paramount to fostering trust and enabling responsible adoption of machine learning technologies in mental health care settings.

1.3 PROBLEM STATEMENT

Mental health disorders, including depression, anxiety, and stress-related conditions, have become increasingly widespread across the globe, affecting millions of people regardless of age, gender, or socioeconomic status. Despite their growing prevalence, these disorders often remain undiagnosed and untreated. This gap in mental health care is largely driven by multiple factors, such as social stigma, which discourages individuals from openly discussing their struggles or seeking help. Furthermore, a lack of awareness about mental health symptoms and available resources contributes to delayed diagnosis.

Additionally, access to professional mental health care remains limited, especially in underserved and remote areas, where there may be a shortage of trained clinicians or mental health facilities. Traditional diagnostic approaches typically rely on self-reporting through interviews or questionnaires, combined with clinical evaluation by mental health professionals. While valuable, these methods can be subjective, time-intensive, and sometimes inaccessible due to geographic, financial, or cultural barriers. The reliance on personal disclosure also introduces variability and potential inaccuracies, making it challenging to identify individuals in the early stages of mental illness when intervention is most effective.

In response to these challenges, there is a pressing need to develop scalable, objective, and efficient methods for early detection and monitoring of mental health conditions. Advances in technology, particularly in the field of machine learning, offer promising avenues to bridge this gap. Machine learning algorithms can analyze large volumes of complex data from diverse sources, identifying subtle patterns and correlations that may be invisible to human clinicians.

By harnessing these capabilities, it becomes possible to create tools that support early diagnosis, continuous monitoring, and personalized treatment recommendations. Such tools have the potential to reach a wider population, including those reluctant or unable to seek traditional care, thereby reducing the burden of untreated mental health disorders on individuals and society.

It aims to contribute to this growing field by developing a machine learning-based system designed to detect signs of mental health issues from multiple types of data. These data sources may include textual content from social media platforms, which can reveal users' mood, thoughts, and social interactions; sensor data from wearable devices.

1.4 OBJECTIVE

Student Mental Health Detection using machine Learning project aims to develop an accurate, machine learning-based predictive model to identify mental health conditions such as anxiety, depression, and panic attacks among students. Leveraging student mental health data, the study focuses on using Decision Tree and Random Forest classifiers to detect individuals who may require psychological support. The primary objective is to create a robust and reliable system that can assist in the early detection of mental health risks, enabling timely intervention and support.

To achieve this aim, several specific objectives are addressed throughout the project. First, the collected student mental health data is analyzed to identify key risk factors and select the most relevant features that significantly influence mental health outcomes. This step involves cleaning, normalizing, and balancing the dataset to ensure fairness and improve model performance.

Next, feature engineering is conducted to extract meaningful attributes from the data, including mental health history, behavioral trends, and environmental triggers, which contribute to more accurate predictions. The project then focuses on developing and training predictive models using both Decision Tree and Random Forest algorithms.

1.5 IMPLMENTATION

The implementation of this project involves several systematic steps to develop a machine learning-based model for predicting mental health conditions such as anxiety, depression, and panic attacks among students. Initially, student mental health data is collected from sources such as questionnaires, clinical assessments, or self-reported surveys. The raw data is then preprocessed to handle missing values, normalize numerical features, and encode categorical variables, ensuring it is clean, balanced, and suitable for machine learning algorithms.

In implementation, feature selection techniques are applied to identify the most relevant attributes that significantly influence mental health outcomes, such as academic stress, social relationships, sleep patterns, or prior mental health history.

After preparing the dataset, two machine learning models Decision Tree and Random Forest classifiers are developed. These models are trained on the labeled dataset, where each instance is associated with a specific mental health condition or risk level. The training process includes hyperparameter tuning using cross-validation to optimize model performance and prevent overfitting.

CHAPTER 2
LITERATURE SURVEY

2.1 PREDICTION OF MENTAL HEALTH PROBLEMS AMONG CHILDREN USING MACHINE LAEARNING

AUTHOR: Ms.Sumathi And Dr.B.Poorna

YEAR OF PUBLICATION : 2016

The paper "Prediction of Mental Health Problems Among Children Using Machine Learning" investigates the use of predictive algorithms to identify early signs of psychological disorders in children. The study acknowledges the growing prevalence of mental health issues in younger populations and the urgent need for scalable, data-driven tools for early detection.

2.2. JUDGING MENTAL HEALTH DISORDERS USING THE DECISION TREE MODELS

AUTHOR: Sandip Roy, P.S. Aithal, Rajesh Bose

YEAROFPUBLICATION:2017

The paper ″Judging Mental Health Disorders Using the Decision Tree Models" explores the application of decision tree-based machine learning techniques for the classification and diagnosis of mental health disorders. The authors focus on the use of structured datasets derived from surveys or clinical records that contain features related to psychological, demographic, and behavioral indicators.

2.3 APPLICATION OF MACHINE LEARNING METHODS IN MENTAL HEALTH DETECTION

AUTHOR: Rohizah AbdRahman, Khairuddin.,

YEAR OF PUBLICATION: 2020

The paper ″Application of Machine Learning Methods in Mental Health Detection" explores the growing role of artificial intelligence in identifying and assessing mental health conditions through automated data analysis. The authors begin by highlighting the limitations of traditional mental health diagnostics, which often rely on self-reported symptoms and time-intensive clinical evaluations.

2.4 A MACHINE LEARNING ALGORITHM TO DIFFERENTIATE BIPOLAR DISORDER FROM DEPRESSIVE DISORDER USING AN ONLINE MENTAL HEALTH QUESTIONNAIRE AND BLOOD BIOMARKER DATA

AUTHOR : Jakub Tomasik, Sung Yeon Sarah Han, Jason D. Cooper.,

YEAR OF PUBLICATION : 2021

The paper ″A Machine Learning Algorithm to Differentiate Bipolar Disorder from Depressive Disorder Using an Online Mental Health Questionnaire and Blood Biomarker Data" addresses a significant challenge in psychiatric diagnostics: the accurate differentiation between bipolar disorder and major depressive disorder. These two conditions often present with similar depressive symptoms, making clinical distinction difficult without extensive evaluation.

2.5 PREDICTING MENTAL HEALTH ILLNESS USING MACHINE LEARNING

AUTHOR: Konda Vaishnavi, U Nikhitha Kamath, B Ashwath Rao and N V Subba Reddy

YEAR OF PUBLICATION : 2021

The paper "Predicting Mental Health Illness Using Machine Learning" explores the application of artificial intelligence techniques to forecast mental health conditions based on patterns found in personal, behavioral, and demographic data. Recognizing the growing mental health crisis and the limitations of traditional diagnostic methods, the authors aim to automate the early detection process using predictive algorithms.

2.6 MENTAL HEALTH PREDICTION MODELS USING MACHINE LEARNING IN HIGHER EDUCATION INSTITUTION

AUTHORS: Sofianita Mutalib et al.

YEAR OF PUBLICATION : 2021

The paper titled "Mental Health Prediction Models Using Machine Learning in Higher Education Institution" by Sofianita Mutalib, Nurul Aiman Zakaria, and Nurul Izzati Jamaluddin explores the application of machine learning techniques to predict mental health issues among university students. Published in 2021 in the Turkish Journal of Computer and Mathematics Education, the study addresses the growing concern of mental health challenges in academic environments.

2.7 MENTALBERT: PUBLICLY AVAILABLE PRETRAINED LANGUAGE MODELS FOR MENTAL HEALTHCARE

AUTHORS: Shaoxiong Ji, Tianlin Zhang, Luna Ansari, Jie Fu, Prayag Tiwari, Erik Cambria

YEAR OF PUBLICATION : 2021

The paper titled "MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare", authored by Shaoxiong Ji, Tianlin Zhang, Luna Ansari, Jie Fu, Prayag Tiwari, and Erik Cambria, presents the development of domain-specific language models tailored for mental health applications. Published in 2021, this research introduces MentalBERT, a variant of BERT (Bidirectional Encoder Representations from Transformers).

2.8 MENTAL ILLNESS CLASSIFICATION ON SOCIAL MEDIA TEXTS USING DEEP LEARNING AND TRANSFER LEARNING

AUTHORS: Iqra Ameer, Muhammad Arif, Grigori Sidorov, Helena Gómez-Adorno, Alexander Gelbukh

YEAR OF PUBLICATION : 2022

The paper "Mental Illness Classification on Social Media Texts Using Deep Learning and Transfer Learning" investigates how advanced natural language processing (NLP) techniques can be utilized to detect signs of mental illness through users' social media activity. With the rise in mental health-related content being shared online, especially on platforms like Twitter, Reddit, and Facebook.

2.9 AN EMPIRICAL COMPARISON OF MACHINE LEARNING MODELS FOR STUDENT'S MENTAL HEALTH ILLNESS ASSESSMENT

AUTHORS: Prathamesh Muzumdar, Ganga Prasad Basyal, Piyush Vyas

YEAR OF PUBLICATION : 2022

The paper "An Empirical Comparison of Machine Learning Models for Student's Mental Health Illness Assessment" explores the effectiveness of various machine learning algorithms in detecting and assessing mental health issues among students. Recognizing the rising prevalence of mental health problems within educational settings, the study aims to identify which computational models provide the most accurate and reliable predictions based on student data collected through surveys and behavioral records.

2.10 MACHINE LEARNING ALGORITHMS FOR DEPRESSION: DIAGNOSIS, INSIGHTS, AND RESEARCH DIRECTIONS

YEAR: 2022

AUTHORS: Shumaila Aleem, Noor ul Huda, Rashid Amin, Samina Khalid, Sultan S. Alshamrani, Abdullah Alshehri

The paper "Machine Learning Algorithms for Depression: Diagnosis, Insights, and Research Directions" presents a comprehensive overview of how machine learning techniques are revolutionizing the diagnosis and understanding of depression. It highlights the challenges of traditional diagnostic methods, which often rely heavily on subjective clinical evaluations and patient self-reporting.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The current landscape of mental health prediction systems reflects a significant shift toward the integration of technology in healthcare. With the increasing awareness of mental health issues and their impact on individuals and society, there is a rising demand for tools that can assist in early detection and intervention. Technological advancements are playing a crucial role in transforming how mental health conditions are understood, monitored, and managed, making the prediction of such conditions more accessible and data-driven.

A wide range of methodologies and approaches are employed in existing systems to predict mental health conditions. These include traditional statistical models such as linear regression, logistic regression, and time-series analysis, which offer interpretable insights but may have limitations in capturing complex patterns. These methods are often used in clinical research and basic assessments, relying heavily on predefined criteria and assumptions about the data.

More recently, there has been a strong shift toward advanced machine learning techniques, which offer greater flexibility and accuracy in handling large, diverse datasets. Algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks are increasingly used to detect patterns in behavioral, demographic, and psychological data. These models can learn from real-world inputs to predict mental health conditions such as anxiety, depression, and stress, contributing to more personalized and proactive mental healthcare solutions.

3.1.1 TRADITIONAL APPROACHES

Traditional mental health assessment methods typically involve the use of standardized questionnaires, clinical interviews, and manual evaluations conducted by trained healthcare professionals. These tools are designed to evaluate symptoms, emotional states, and behavioral patterns in patients to identify potential mental health conditions. While these methods have been foundational in clinical psychology and psychiatry for decades, they often require significant time and resources, both from practitioners and patients.

Despite their clinical value, these methods are often subjective and dependent on the expertise and interpretation of the healthcare provider. The outcome of an assessment can vary based on the professional's background, experience, and interaction with the patient. This subjectivity introduces variability, which can affect the consistency and reliability of diagnoses. Moreover, patients may sometimes underreport or miscommunicate their symptoms due to stigma, discomfort, or lack of awareness, further complicating accurate assessment.

Another key limitation of traditional methods is their inability to detect complex and subtle patterns within patient data. These methods generally rely on direct responses or observable behavior, which might overlook underlying trends or correlations that aren't immediately apparent. As a result, there can be delays in diagnosis or even misdiagnosis, preventing timely and appropriate intervention. This highlights the need for more advanced, data-driven approaches such as those offered by machine learning to enhance the accuracy and efficiency of mental health assessments.

3.1.2 MACHINE LEARNING-BASED SYSTEMS

With the advancement of technology, the field of mental health assessment has witnessed a notable transition toward machine learning-based systems. These systems are designed to enhance traditional diagnostic methods by automating and improving the prediction of mental health conditions. The growing availability of digital data, combined with improvements in computing power and algorithmic efficiency, has made it possible to develop tools that can analyze vast and complex datasets with high speed and accuracy.

Machine learning-based systems are capable of leveraging large-scale datasets that include diverse information such as patient behavior, clinical history, lifestyle factors, and even social media activity. By processing this data, these systems can identify hidden patterns, trends, and correlations that may not be apparent through manual analysis. This allows for a more comprehensive understanding of an individual's mental health, enabling predictions that are both data-driven and evidence-based.

Incorporating advanced algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks, these systems can model the progression and risk factors of mental health disorders with high precision. They are not only useful for diagnosing existing conditions but also for predicting potential future mental health challenges, allowing for early intervention and personalized care plans. As a result, machine learning has become an essential tool in the evolution of mental health care, offering improved accuracy, scalability, and accessibility compared to traditional methods.

3.2 PROPOSED SYSTEM

The proposed method is to build a ML model to predict mental health (depression). The data set is first pre-processed and the columns are analyzed and then different machine learning algorithms would be compared to obtain the predictive model with maximum accuracy. Data is loaded, checked for cleanliness, and then trimmed and cleaned for analysis. The data set collected for predicting given data is split into Training set and Test set. The Data Model which was created using machine learning algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

ML algorithms prediction model is effective because of the following reasons :It provides better results in classification problem. It is strong in pre-processing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables. It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with. These reports are to the investigation of applicability of machine learning techniques for Mental Health prediction in operational conditions. Finally, it highlights some observations on future research issues, challenges, and needs.

The system will employ machine learning algorithms, particularly:

Decision Tree: For interpretable and straight forward classification.

Random Forest: To improve prediction accuracy by aggregating the outputs of multiple decision trees.

3.3 BLOCK DIAGRAM OF PROPOSED SYSTEM

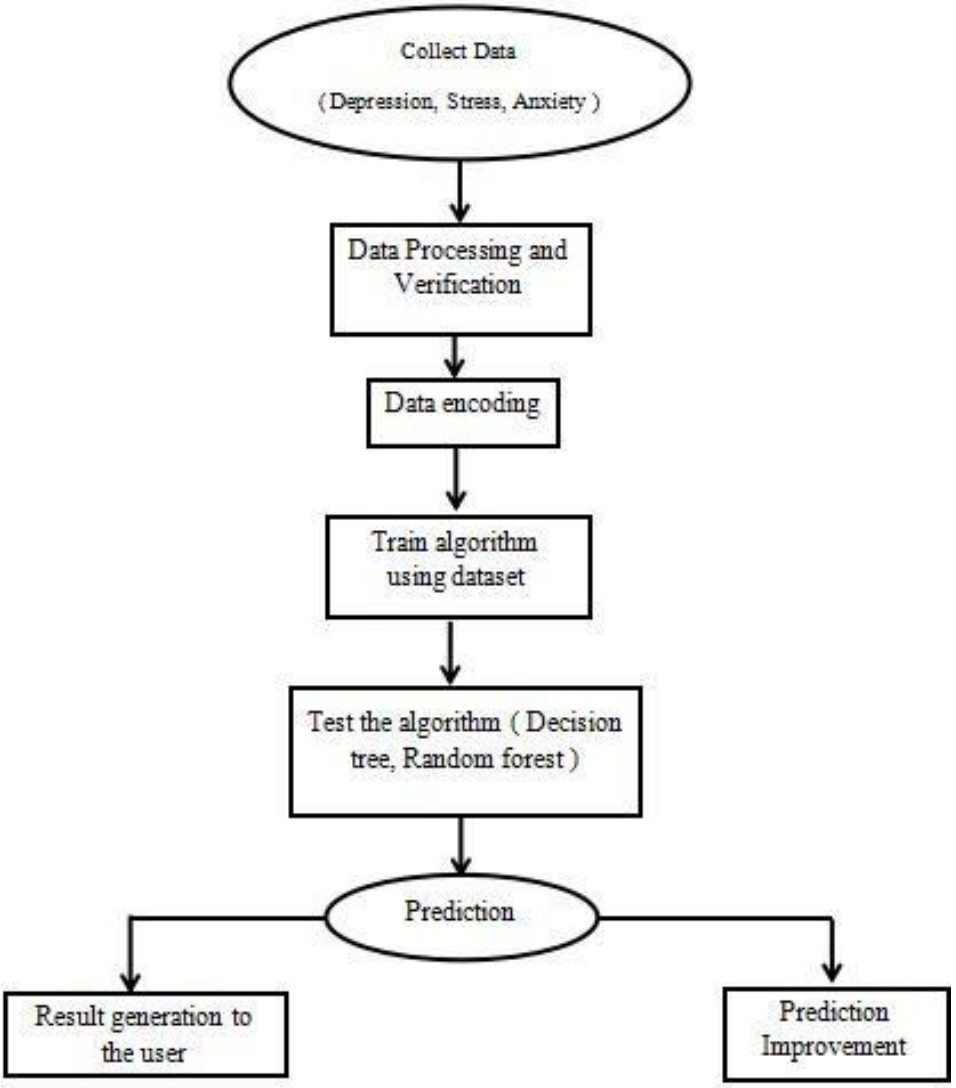


Fig.3.3 Proposed System of Mental Detection

CHAPTER 4

MODULES

4.1 MODULE DESCRIPTION

- Data Exploration
- Data Cleaning
- Training and Validation
- Model Selection & Optimization Module
- Testing

4.1.1 Data Exploration

Predict mental health outcomes using machine learning algorithms like Random Forest and Decision Tree, the process starts with exploring the dataset to understand its structure. Key statistics like mean and median are calculated, and visualizations such as histograms and heatmaps help identify trends, outliers, and missing values. Data preprocessing follows, addressing missing values, encoding categorical variables, and splitting the data into training and testing sets. Scaling may be applied but is less critical for tree-based models.

The Decision Tree algorithm splits data into decision points to capture patterns, while Random Forest builds multiple trees and combines their predictions to improve accuracy and reduce overfiting. After training, predictions are evaluated using metrics like accuracy, precision, and recall, with confusion matrices providing further insights. Random Forest also highlights feature importance, identifying factors most influencing mental health outcomes. Decision Trees are simple but prone to overfitting.

4.1.2 Data Cleaning

Data cleaning is essential for predicting mental health outcomes using algorithms like Random Forest and Decision Tree. It ensures the dataset is accurate, consistent, and reliable. Missing values, common in mental health data, can be handled by filling them with the mean, median, or mode, or by removing rows or columns with too many gaps. Duplicate records should be removed to avoid bias, and invalid or inconsistent data, like incorrect age values, must be fixed or excluded. Standardizing formats, such as ensuring dates are consistent and text labels are uniform, further improves data quality.Outliers, which can distort predictions, should be identified and addressed using box plots or statistical methods. Categorical variables, like gender or employment status, need to be converted into numerical forms using techniques like one-hot or label encoding. Class imbalance, where one outcome is more frequent than another, can be corrected using methods like oversampling, undersampling, or synthetic techniques. While Random Forest and Decision Tree are less affected by unscaled data, proper preprocessing reduces noise and boosts model performance. Clean data enhances Random Forest's ability to combine accurate predictions across trees and helps Decision Trees generalize better, leading to more reliable predictions and actionable insights.

4.1.3 Training and Validation

In this module, the dataset is split into training and validation sets to build and fine-tune the models. The Decision Tree and Random Forest algorithms are trained on the training set to classify mental health conditions based on input features such as demographic details, behavioral patterns, and responses to mental health questionnaires.

Validation is performed by testing the models on a separate validation set to measure their generalization performance. Metrics such as accuracy, precision, recall, and F1-score are calculated to evaluate the models. Cross-validation techniques like k-fold cross-validation are used to ensure the stability and reliability of the models across multiple splits of the datasets.

4.1.4 Model Selection & Optimization Module

The Model Selection & Optimization module plays a vital role in enhancing the accuracy and reliability of mental health prediction. In this project, the module focuses on evaluating and comparing two supervised machine learning algorithms decision tree classifiers and random forest algorithms to determine the most effective model for early detection of mental health issues. Initially, the dataset is split into training and testing subsets using techniques such as k-fold cross-validation to ensure consistent and unbiased model evaluation. Each algorithm is trained using the same set of features, and hyperparameter tuning is performed through grid search and random search methods to optimize model parameters such as tree depth, the number of estimators, and the minimum number of samples per split. These optimizations help improve generalization and reduce the risk of overfitting, especially in complex or noisy datasets.

To assess performance, multiple evaluation metrics are employed, including accuracy, precision, recall, F1-score. These metrics provide a comprehensive view of how well each model identifies mental health conditions and supports early intervention. The decision tree model is valued for its simplicity and interpretability but may suffer from overfitting in high-dimensional data.

4.1.5 Testing

The testing module uses the models trained in the previous module to evaluate their performance on a separate test dataset, which contains data not seen during training. The models classify mental health conditions based on the input features and generate predictions for each test instance. Performance metrics such as accuracy and confusion matrix are calculated to assess the effectiveness of the models. The predictions are further processed to display the likelihood of mental health issues in percentage form, providing users with clear and interpretable results. The best-performing model, based on testing metrics, is finalized for deployment in the web interface, ensuring reliable real-time predictions for users.

Fig 4.2.4.Training Model Chart

CHAPTER 5
SOFTWARE DESCRIPTION

5.1 HARDWARE SYSTEM REQUIREMENTS

- Processor : Intel Core i3
- RAM : 4 GB
- Hard Disk : 10 GB
- Compact Disk : 650 MB
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor
- Internet : Required for Google Co-lab access

5.2 SOFTWARE SYSTEM REQUIREMENTS

- Operating System : Platform-independent (accessible via any OS with internet, e.g., Windows, macOS, or Linux)
- Development Environment: Google Co-lab (cloud-based Jupyter Notebook)Programming Language: Python
- Libraries and Frameworks: NumPy, Pandas , Matplotlib, Seaborn

5.3 SOFTWARE ENVIRONMENT

5.3.1 SOFTWARE USED

Python is a high-level, versatile, and widely used programming language, particularly in the field of data science and machine learning. It is known for its simplicity and readability, making it an ideal choice for implementing machine learning projects, including your Mental Health Prediction Using ML Algorithms (Decision Tree and Random Forest) project.

5.3.2 PYTHON TECHNOLOGY

- **Rich Libraries and Frameworks:** Python has a vast ecosystem of libraries that facilitate data analysis, machine learning, and visualization, which are essential for this project. For example: Pandas Used for data manipulation and preprocessing, handling datasets efficiently. NumPy Provides support for large, multi-dimensional arrays and matrices, which are often used in machine learning algorithms. Matplotlib and Seaborn Used for data visualization, helping you create plots and charts to understand and communicate results effectively.

- **Ease of Learning and Use:** Python's simple and clean syntax makes it easy for developers to learn and use. Even for complex algorithms like Decision Trees and Random Forests, Python offers straightforward implementations and excellent documentation.

- **Extensive Support for Machine Learning:** Python is one of the most popular languages in the machine learning community. It supports a wide range of machine learning frameworks and algorithms.

- **Integration with Google Co-lab:** Python seamlessly integrates with cloud-based platforms like Google Co-lab, which allows you to run code without worrying about local hardware or software configurations. Google Colab provides a free, easy-to-use environment that supports Python and includes pre-installed libraries for machine learning and data analysis.

- **Community and Resources:** Python has a large community of developers and researchers, which means a wealth of resources, tutorials, and forums are available to help solve any issues you encounter. This makes it easier to find support during development.

5.3.3 HOW PYTHON IS USED IN THE PROJECT

In the Mental Health Prediction project, Python plays a crucial role in managing and preparing the data for analysis. The process begins with loading and preprocessing datasets using the Pandas library, which allows for efficient data manipulation and cleaning. This step ensures that the data is in a suitable format for training machine learning models, handling missing values, encoding categorical variables, and standardizing features where necessary.

Once the data is prepared, Python's Scikit-learn library is used to implement the Decision Tree and Random Forest algorithms. These models are trained on the dataset to identify patterns and predict mental health conditions such as anxiety, depression, and stress. Scikit-learn also provides tools for evaluating model performance, enabling comparison between algorithms to determine which one yields better results in terms of accuracy and reliability.

To interpret and present the findings, the project makes use of visualization libraries such as Matplotlib and Seaborn. These tools help in creating informative plots and graphs that display insights from the data and model performance. Additionally, Python is used to save and load trained models, making it easier to reuse them for future predictions or deploy them in real-world applications. This end-to-end functionality makes Python an ideal language for building and deploying machine learning solutions in mental health prediction.

5.3.4 GOOGLE CO-LAB

Google Colab (short for Colaboratory) is a cloud-based interactive development environment (IDE) developed by Google. It is primarily used for writing and executing Python code in a browser-based interface, making it accessible to users without requiring any local setup. Its user-friendly interface and integration with Google Drive allow for easy file management and storage, enabling users to save and access their work from anywhere.

One of the key advantages of Google Colab is its support for projects involving data analysis, machine learning, and artificial intelligence. It comes pre-installed with popular libraries such as NumPy, pandas, scikit-learn, TensorFlow, and Matplotlib, which are essential tools for building and evaluating machine learning models. Colab also offers free access to GPUs and TPUs, significantly improving the speed and efficiency of model training, especially when working with large datasets.

For a machine learning-based project like Mental Health Prediction Using ML Algorithms (specifically Decision Tree and Random Forest), Google Colab proves to be an ideal platform. It not only supports the entire development pipeline from data preprocessing and feature engineering to model training and evaluation but also facilitates collaboration and sharing, allowing team members to work together in real time. These features make Google Colab an efficient and powerful tool for implementing predictive models in mental health research and beyond.

5.3.5 KEY FEATURES OF GOOGLE COLAB

- **Cloud-Based Environment :** Google Colab is hosted in the cloud, meaning you don't need to install any software or worry about local hardware specifications. You can access your project from anywhere, on any device, as long as you have an internet connection.

- **Pre-installed Libraries and Tools :** Google Colab comes with most of the popular Python libraries pre-installed, such as Pandas, NumPy, Matplotlib, and Seaborn, which are essential for your machine learning project. This eliminates the need to install and configure libraries manually, saving time and effort.

- **Jupyter Notebook Interface :** Google Colab is based on the Jupyter notebook interface, which allows you to write and execute code in a step-by- step manner. You can run Python code blocks individually, visualize data, and display results directly within the notebook. This makes it easier to experiment with different machine learning models, such as Decision Tree and Random Forest, and immediately see the results.

- **Collaborative Environment :** One of the most significant advantages of Google Colab is its collaborative features. You can share your Colab notebooks with others, allowing them to view or edit your work. This is helpful if you're working on a team or need feedback from mentors or colleagues. It also integrates well with Google Drive for easy file sharing and saving.

- **Integration with Google Drive :** Google Colab seamlessly integrates with Google Drive, enabling you to save your datasets, models, and notebooks directly in the cloud. This integration also makes it easy to load data into Colab from your Drive and store trained models for future use.

- **Support for Python Code and Machine Learning :** Since your project is focused on implementing machine learning models (Decision Tree and Random Forest), Colab is an ideal platform. It supports Python's extensive machine learning libraries and offers a smooth experience for implementing, training, and evaluating models.

- **Easy Visualization:** Google Colab supports rich text formatting, including charts, graphs, and plots generated by libraries such as Matplotlib and Seaborn. Visualizing results is an essential part of understanding the performance of your models and presenting your findings clearly.

- **No Setup Required:** Unlike local environments, Google Colab requires no setup on your machine. You can begin working on your project immediately by simply signing in with your Google account. This is particularly useful when experimenting with various models or datasets without worrying about local configuration issues.

5.3.6 HOW GOOGLE COLAB IS USED IN THIS PROJECT

For the Mental Health Prediction Using ML Algorithms project, Google Colab is used in the following ways:

- **Data Loading and Preprocessing:** You can upload datasets to Google Drive or directly into Colab, then use Pandas to clean and preprocess the data.

- **Model Implementation**: Google Colab allows you to easily implement machine learning algorithms like Decision Trees and Random Forest .You can train and evaluate these models directly within the notebook.

- **Model Training and Evaluation:** With access to GPUs and TPUs, Colab accelerates model training, making it quicker to test different configurations and improve performance.

- **Data Visualization**: Use Matplotlib and Seaborn to create graphs and visualizations that help interpret the results of the models, such as accuracy, confusion matrices, or feature importance. \

- **Collaborative Sharing:** You can share the notebook with team members, mentors, or stakeholders to review or collaborate on the project.

5.3.7 ADVANTAGES OF USING GOOGLE COLAB

One of the major advantages of Google Colab is its cost-effectiveness. It is completely free to use and provides access to powerful cloud-based computational resources, including GPUs and TPUs, without the need for any local hardware setup. This makes it an ideal platform for students, researchers, and developers working on machine learning projects without access to expensive equipment.

Another key benefit is its ease of use. Google Colab adopts a notebook-style interface that allows users to write code, display results, add explanations, and document the entire workflow—all in a single environment. This format promotes better organization, readability, and understanding, making it especially useful for collaborative projects and educational purposes.

Google Colab also excels in terms of scalability. It can handle large datasets and complex machine learning models effectively, thanks to its access to high-performance hardware and cloud infrastructure. Whether you're building a simple prototype or training deep learning models, Colab’s scalability ensures that your development process remains smooth and efficient as your project grows in complexity.

5.3.8 PYTHON LIBRARIES

Pandas is used to load, clean, and preprocess the dataset. It provides efficient data structures like DataFrames for handling tabular data (rows and columns). You can read data from CSV, Excel, or SQL databases, clean missing values, and filter or transform the data before applying machine learning models.

NumPy is used to handle numerical operations, such as array manipulation and mathematical computations. Many machine learning algorithms, including Decision Trees and Random Forests, require mathematical operations that NumPy performs efficiently.

Matplotlib is used for creating static, animated, and interactive visualizations. It helps in visualizing the model's performance and the data. Seaborn builds on Matplotlib and provides a higher-level interface for creating more attractive and informative statistical graphics. It's useful for visualizing relationships between features, distributions of data, and model evaluation results.

Together, these libraries provide the necessary tools for data preprocessing, implementing machine learning models, evaluating performance, and visualizing results, making Python the ideal choice for your mental health prediction project using machine learning.

CHAPTER 6
SYSTEM DESIGN

6.1 DATAFLOW DIAGRAM

The data flow diagram (DFD) of the student mental health detection system illustrates the overall flow of data throughout the application, from data collection to prediction output. The process begins with the student filling out a mental health questionnaire, which serves as the primary input source. This data, along with optional physiological or academic records if available, is directed to a preprocessing unit where missing values are handled, and the data is cleaned and normalized. The processed data is then fed into a trained machine learning model such as Decision Tree, Random Forest, or Support Vector Machine—which analyzes the patterns and predicts the likelihood of mental health issues such as stress, anxiety, or depression.

The prediction results are passed to the output module, which generates a report or alert for counselors or mental health professionals. This ensures timely intervention. Throughout the process, a data storage component is involved to securely store input data, processed data, and prediction outcomes. The DFD clearly outlines the interaction between users, processes, data stores, and the prediction engine, ensuring an organized and systematic approach to mental health analysis.

The system also includes an authentication and user management module that ensures only authorized users such as students, counselors, or system administrators can access specific features of the application. Students log in to fill out questionnaires, while counselors access the prediction results and reports. This interaction is reflected in the DFD through clear data flows between the user interfaces and the authentication process.

Additionally, logs and user activity data can be stored and analyzed to improve system performance and ensure compliance with data privacy standards, such as consent and secure storage protocols.

Furthermore, feedback loops are integrated into the system to enhance the machine learning model over time. After predictions are made and interventions are delivered, outcomes or counselor evaluations can be fed back into the system. This feedback is stored and later used during retraining phases to improve the accuracy and relevance of the prediction models. This dynamic flow of data from input collection, model prediction, report generation, and feedback is what makes the DFD a crucial planning tool.

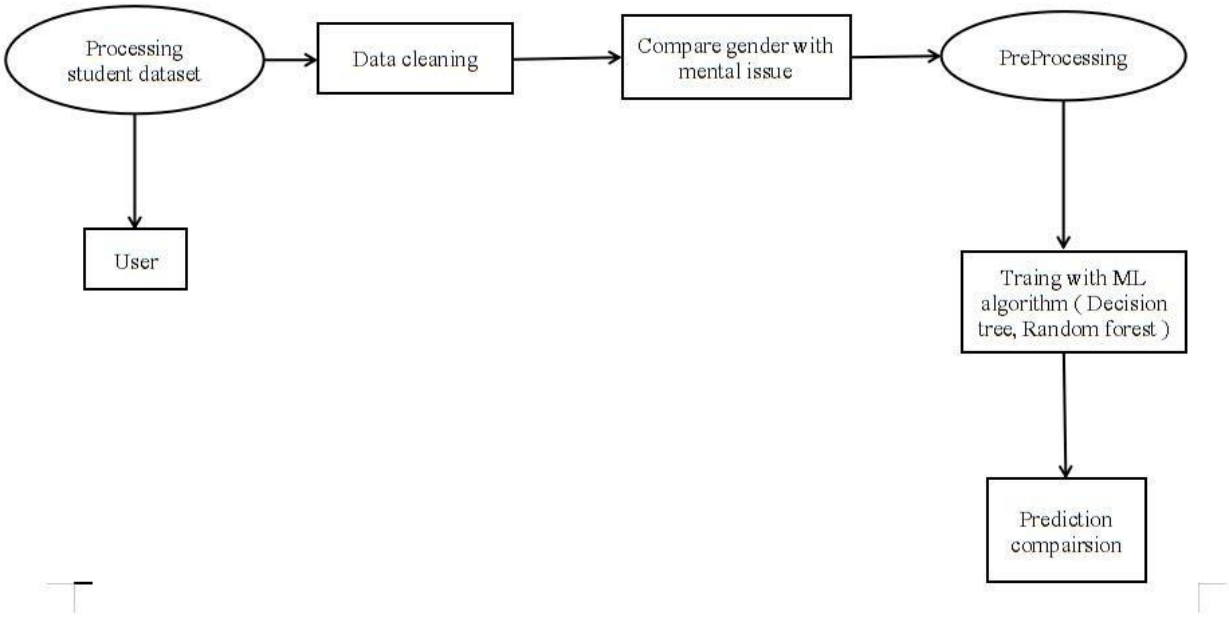


Fig 6.1 Dataflow Diagram

6.2 USECASE DIAGRAM

The use case diagram for the student mental health detection system visually represents the functional interactions between the system and its users. The primary actors in this system include the Student, Counselor (Mental Health Professional), and System Administrator. Each actor interacts with the system through specific use cases. The Student can register, log in, complete a mental health questionnaire, and view their mental health status or recommendations generated by the machine learning model. The Counselor can log in, access student mental health reports, monitor flagged cases, and provide follow-up feedback or recommendations based on the system's output. Meanwhile, the System Administrator manages user accounts, maintains the database, and oversees the machine learning model's training and updates. The use case diagram outlines these interactions clearly, showing how each user role engages with system functions and how those functions contribute to the early detection and monitoring of mental health issues among students.

An important feature highlighted in the use case diagram is the "Predict Mental Health Status" function, which is central to the system. This use case is triggered when a student submits a completed questionnaire. The system processes the data and uses a trained machine learning model to generate a prediction such as identifying signs of stress, anxiety, or depression. This prediction is then stored and made accessible to both the student and the counselor.

The diagram also includes "Provide Feedback", a use case for counselors to record their observations or confirm the prediction outcome. This feedback loop helps improve model accuracy over time and ensures that the predictions are clinically relevant.

The use case diagram reflects non-functional yet essential processes such as “Maintain Data Security” and “Update ML Model”, typically handled by the system administrator. These use cases ensure that user data is encrypted, stored securely, and only accessible to authorized personnel, addressing privacy and ethical concerns. The “Update ML Model” use case allows the system administrator to retrain or fine-tune the machine learning algorithm as new data becomes available, ensuring that the prediction system evolves and adapts to changing student behavior patterns. Together, these use cases form a comprehensive overview of system functionality, clarifying the responsibilities of each user type and contributing to a reliable and responsive mental health support system for students.

The Student can register, log in, complete a mental health questionnaire, and view their mental health status or recommendations generated by the machine learning model. The Counselor can log in, access student mental health reports, monitor flagged cases, and provide follow-up feedback or recommendations based on the system’s output. Meanwhile, the System Administrator manages user accounts, maintains the database, and oversees the machine learning model's training and updates. The use case diagram outlines these interactions clearly, showing how each user role engages with system functions and how those functions contribute to the early detection and monitoring of mental health issues among students.

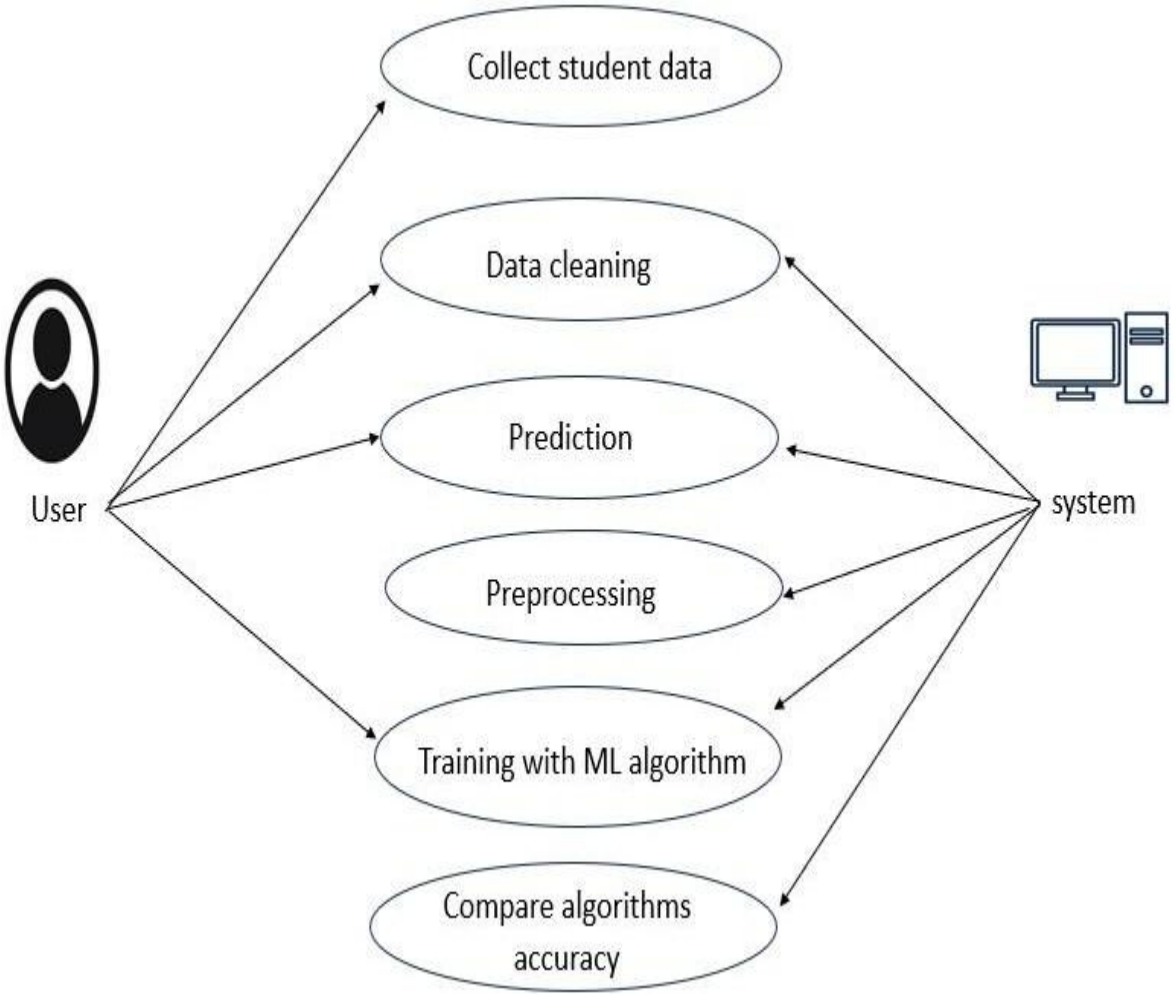


Fig 6.2 Usecase Diagram

6.3 CLASS DIAGRAM

The model class structure is a fundamental aspect of system design that organizes and represents the various components of a software application using design elements such as classes, packages, and objects. These elements serve as the building blocks of object-oriented design, enabling developers to conceptualize, specify, and implement complex systems in a structured and manageable way. Classes, in particular, play a central role in this structure by encapsulating the data and behavior of entities within the system. A class defines the blueprint for objects, specifying the properties (attributes) that characterize the object and the operations (methods) that define its behavior. This encapsulation facilitates modularity, reusability, and maintainability, which are critical qualities for scalable software development.

Class diagrams are an essential tool used to visually represent the static structure of a system at different stages of its design. They provide a comprehensive view of the system from multiple perspectives, including the conceptual, specification, and implementation levels. The conceptual perspective focuses on modeling the real-world entities and their relationships that the system needs to handle, helping stakeholders understand the domain without delving into technical details. The specification perspective refines this understanding by defining the precise attributes and operations of each class, laying the groundwork for system functionality. Finally, the implementation perspective deals with how these classes and their interactions will be realized in actual code, including details related to data types, access modifiers, and inheritance hierarchies. By capturing these perspectives, class diagrams bridge the gap between abstract system requirements and concrete software solutions.

Classes within these diagrams are composed of three primary components: the class name, attributes, and operations. The class name uniquely identifies the class, typically reflecting the concept or entity it represents. Attributes represent the data members or properties associated with the class, describing the state or characteristics of the objects instantiated from the class. Operations, on the other hand, define the functions or methods that the class can perform, outlining the behavior and capabilities of the class instances.

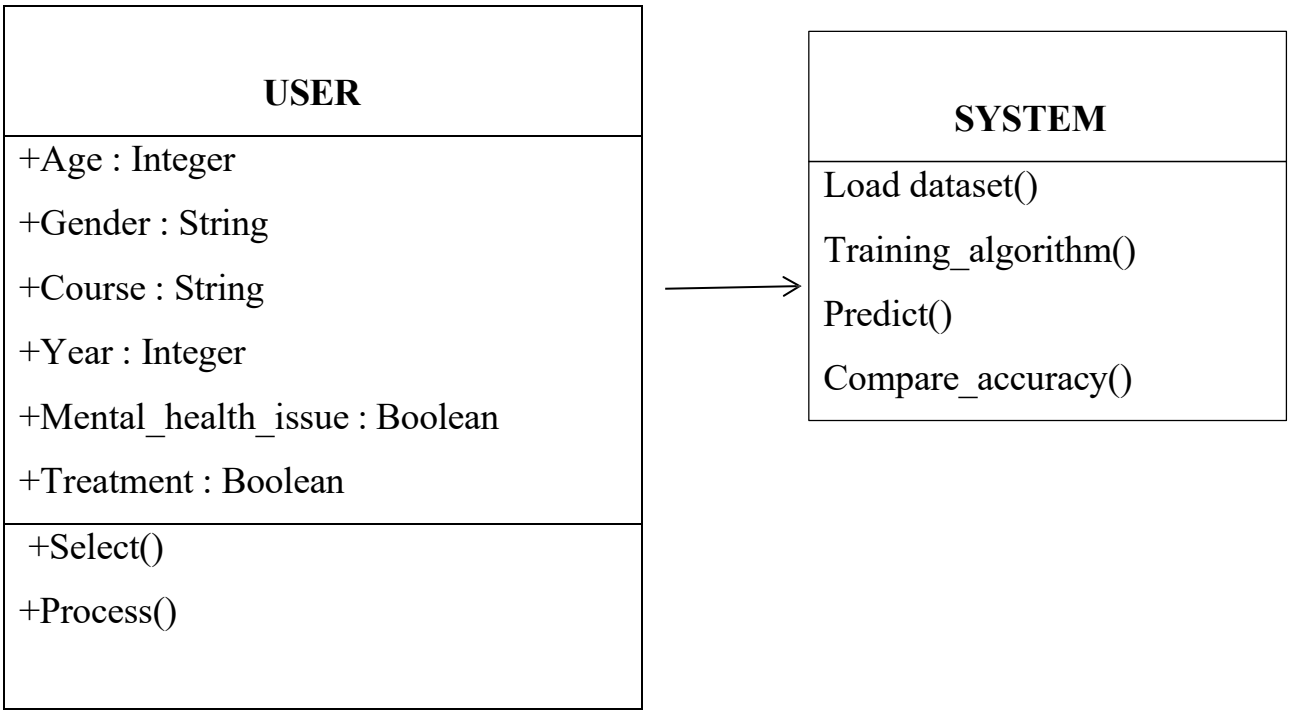


Fig 6.3 Class Diagram

6.4 SEQUENCE DIAGRAM

Sequence diagrams are a crucial type of interaction diagram used in software engineering and system design to visually represent the flow of time-based interactions between various objects involved in a particular process or scenario. These diagrams display the sequence in which objects communicate with each other by exchanging messages, focusing on the chronological order of events that occur during the execution of a system function. The diagram is organized along two main dimensions: the vertical axis represents time progression, moving downward to show the sequence of interactions as they happen, while the horizontal axis depicts the different objects or participants involved in the interaction. This clear representation of the temporal sequence helps designers and developers understand how system components collaborate over time to achieve specific goals or use case functionalities.

In sequence diagrams, objects are considered as entities that exist at particular points in time and have a unique identity that distinguishes them from other objects. Each object has a state and a set of attributes that define its value at that moment. The diagram captures these objects as lifelines that extend vertically along the time axis, illustrating the duration over which an object participates in the interaction. Messages between objects are shown as arrows pointing from the sender to the receiver, indicating communication such as method calls, responses, or signals. The arrows are arranged sequentially according to when the messages occur, which provides a clear, step-by-step view of the interaction flow. This visualization is especially helpful for understanding complex scenarios where multiple objects interact in intricate patterns to perform a task or respond to an event.

Sequence diagrams are tightly linked to the logical view of the system under development, where they are often used to realize use cases by detailing the dynamic behavior of objects during particular interactions. They serve as practical tools for bridging the gap between high-level requirements captured in use cases and the detailed design needed for implementation. Sometimes, sequence diagrams are also referred to as event diagrams or event scenarios, emphasizing their role in modeling the response of a system to events and the consequent interactions. By illustrating the objects involved, the messages exchanged, and the order of these exchanges, sequence diagrams provide a comprehensive and intuitive understanding of how the system operates in real time, enabling stakeholders to validate system behavior, identify potential design flaws, and communicate effectively among development teams.

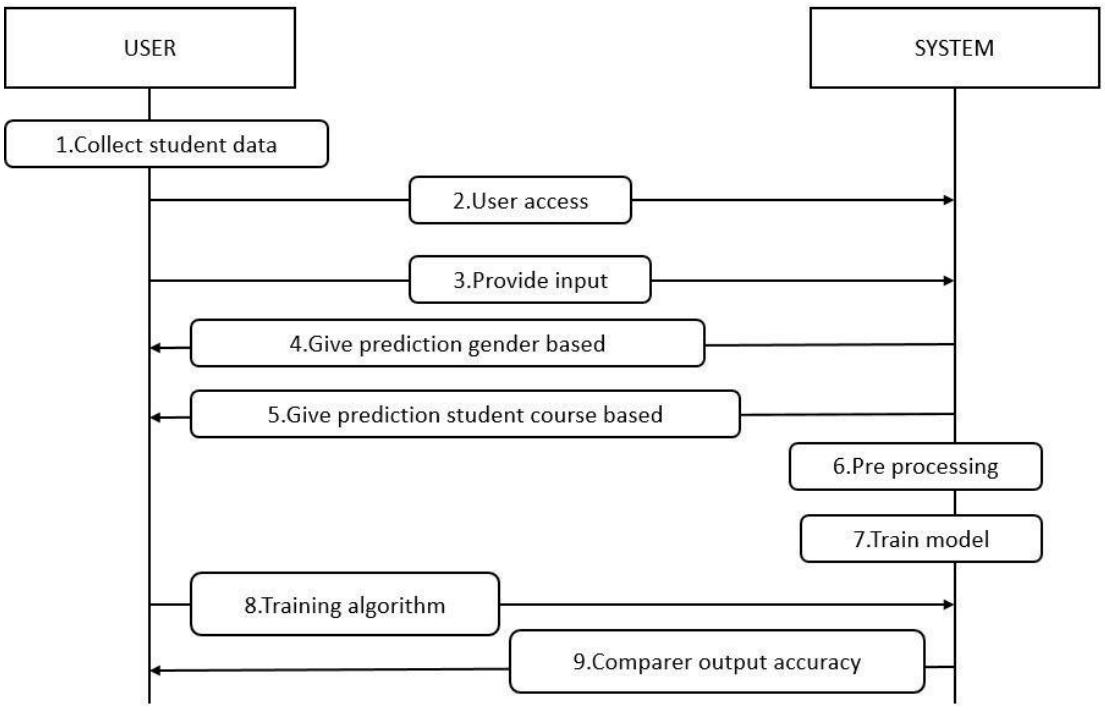


Fig 6.4 Sequence Diagram

6.5 STATE CHART DIAGRAM

A State Chart Diagram is a type of Unified Modeling Language diagram that represents the state machine of a system, showcasing the behavior of classes over time. It focuses on how objects behave rather than what triggers their actions. This diagram models the dynamic behavior of objects by outlining their transitions from one state to another throughout their lifecycle.

Unlike other UML diagrams that may focus on operations or processes, a State Chart Diagram does not emphasize the commands or methods that cause changes. Instead, it highlights the actual changes in state as objects evolve. It is especially useful for understanding how objects respond to various events, and how those events cause shifts between defined states.

One of the primary purposes of this diagram is to illustrate how an object changes from one state to another. These changes are usually triggered by external or internal events. The life cycle of each object in a class can be fully understood by analyzing its state transitions, making this diagram a valuable tool in software design and modeling.

In a State Chart Diagram, there are generally two key states: the Initial State, which marks where the object begins its lifecycle, and the Final State, where the object's lifecycle ends. Other essential components include the State itself, which represents a specific condition or situation of an object. During a state, the object may perform certain actions, satisfy specific conditions, or wait for certain events to occur before transitioning to the next state.

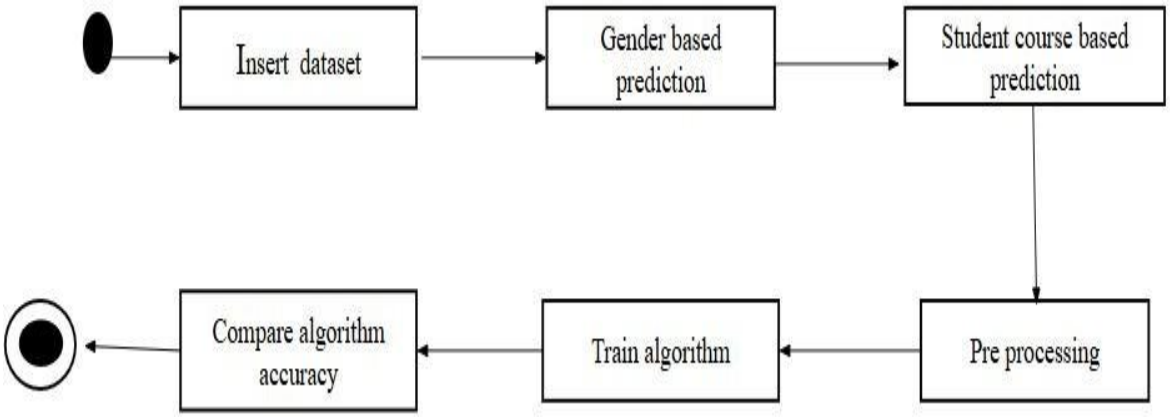


Fig 6.5 State chart diagram

CHAPTER 7
RESULT AND DISCUSSION

7.1 RESULT

Test each step in the preprocessing pipeline (e.g., handling missing values, data normalization, and encoding categorical variables) to ensure that data is correctly prepared for model training. Validate that the most relevant features, such as symptoms of anxiety, depression, and stress, are accurately identified and used in the models, ensuring optimized model performance. Evaluate the Decision Tree and Random Forest algorithms separately, testing their ability to correctly classify mental health conditions using sample datasets. Ensure each algorithm performs as expected and generates appropriate metrics.Assess individual model accuracy, precision, recall, and F1 score to confirm each model meets the desired performance standards. Compare results with established benchmarks to gauge the system's effectiveness. Test the system's responses to potential errors, such as missing input data or out-of-range values, ensuring appropriate error messages and recovery mechanisms are in place.

7.2 USER ACCEPTANCE TESTING

End-to-End Functionality: Conduct comprehensive testing of the entire system, from data input to mental health prediction output, using real-world datasets. Evaluate the system's overall functionality, ensuring predictions align with clinical insights. Test the speed and accuracy of predictions across various testing conditions. Assess model performance under diverse data inputs, ensuring predictions remain accurate and timely. Ensure the user interface is intuitive and accessible, allowing healthcare professionals and users to interact effectively with the system.

7.3 CONCLUSION

Student Mental Health Detection using Machine Learning project explores the capabilities of machine learning algorithms, specifically Decision Tree and Random Forest classifiers, in predicting mental health conditions such as anxiety, depression, and stress. By utilizing real-world survey data from diverse sources, the study demonstrates how predictive models can play a vital role in early detection and intervention for individuals at risk. The use of such data-driven approaches showcases the potential of technology in supporting mental health awareness and improving public health outcomes.

The entire development process was carried out using Google Colab, a powerful cloud-based platform that enabled efficient computation, easy integration of interactive visualizations, and seamless collaboration. The project workflow included critical stages such as data pre-processing, feature engineering, model training, and performance evaluation. A major emphasis was placed on the selection of relevant features—including demographic attributes, lifestyle habits, and psychosocial indicators—which were found to significantly affect mental health predictions.

Among the algorithms used, the Random Forest classifier proved to be particularly effective due to its high accuracy and robustness in handling complex and non-linear feature interactions. It consistently outperformed the Decision Tree model in predictive performance. However, the Decision Tree model also provided considerable value by offering interpretability, which is crucial for understanding the influence of individual factors on mental health outcomes. Together, these models highlight the balance between accuracy and explainability, making them promising tools in the field of mental health prediction.

7.4 FUTURE ENHANCEMENT

The "Mental Health Prediction using Machine Learning Algorithms" project can be enhanced by integrating additional machine learning algorithms to improve prediction accuracy and robustness. Algorithms like Support Vector Machines (SVM), Gradient Boosting, or XG Boost could be explored alongside Decision Trees and Random Forests to compare and find the most efficient model. Additionally, expanding the dataset by including more diverse data points, such as social media activity, sleep patterns, or lifestyle habits, can provide a more comprehensive understanding of mental health.

Incorporating deep learning techniques, particularly Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) networks, can help capture complex temporal patterns in mental health data, further enhancing prediction capabilities. The project can also be extended by creating a more user-interactive website that offers personalized recommendations based on the predictions and includes real-time analysis with visualization tools to make the results more actionable for users. These advancements will ensure the system is more accurate, scalable, and user- friendly, providing deeper insights into mental health prediction.

APPENDIX – A

SOURCE CODE

Initialization

```
!pip install seaborn –U
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns. version
from google.colab import files
uploaded = files.upload()
data = pd.read_csv('/content/Student_Mental_health.csv')
df_clean = data.copy()
df_clean.head(5)
```

Data Cleaning

```
# Check dtypes
print('Col types:\n',df_clean.dtypes,'\n','='*25,sep='')
# Check for NA values
print('Number of NA per Col:')
df_clean.isna().sum()
# Since only age has NA we can replace it with the mean as an int
df_clean['Age'] = df_clean['Age'].fillna(df_clean['Age'].mean()).astype('int64')
df_clean.isna().sum()
# Rename columns for clarity
df_clean.rename(columns={
'Choose your gender':'Gender',
'What is your course?':'Course',
'Your current year of Study':'Year',
```

```
'What is your CGPA?':'GPA',
'Marital status':'Married',
'Do you have Depression?':'Depression',
'Do you have Anxiety?':'Anxiety',
'Do you have Panic attack?':'Panic_Attacks',
'Did you seek any specialist for a treatment?':'Treatment'}, inplace=True)
```

Gender to Responses & Conditions

```
# Compare Gender to Mental Health conditions
sns.countplot(data=df_clean, hue='Anxiety', x='Gender', hue_order=['Yes','No'])
plt.title('Anxiety by Gender')
plt.xlabel('')
plt.show()
sns.countplot(data=df_clean,hue='Depression',x='Gender',
hue_order=['Yes','No'])
plt.title('Depression by Gender')
plt.xlabel('')
plt.show()
sns.countplot(data=df_clean,hue='Panic_Attacks',x='Gender',
hue_order=['Yes','No'])
plt.title('Panic Attacks by Gender')
plt.xlabel('')
plt.show()
```

Course to Conditions with Gender

```
# Compare courses to Anxiety and Gender
plt.figure(figsize=(10, 10))
sns.swarmplot(data=df_clean,x='Anxiety',y='Course',
hue='Gender',order=['Yes','No'])
```

```
plt.show()
# Compare Courses to Depression and Gender
plt.figure(figsize=(10, 10))
sns.swarmplot(data=df_clean,x='Depression',y='Course',
hue='Gender',order=['Yes','No'])
plt.show(
# Compare Courses to Panic Attacks and Gender
plt.figure(figsize=(10, 10))
sns.swarmplot(data=df_clean, x='Panic_Attacks', y='Course',
hue='Gender',order=['Yes','No']) plt show()
```

Modelling

```
for col in df_clean.columns:
print(df_clean[col].value_counts().sort_index(),'\n','='*50,sep='')
```

Data Pre-Processing

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report,accuracy_score
from sklearn.preprocessing import StandardScaler
df_model = df_clean.copy()
df_model.drop(columns='Timestamp',inplace=True)
df_model.dtypes
#Convert Binary columns into numeric
for col in bool_cols:
df_model[col] = df_model[col].replace({'Yes':1,'No':0})
```

Train models

```
# Split data
X = df_model.drop(columns=['Depression'])
y = df_model['Depression']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=9)
# Decision Tree
decision_tree=DecisionTreeClassifier(random_state=43)
decision_tree.fit(X_train,y_train)
print('DecisionTree:',cross_val_score(decision_tree,X_train,y_train,cv=8).mea())
# Random Forest
random_forest = RandomForestClassifier(random_state=43)
random_forest.fit(X_train,y_train)
print('Random Forest:',cross_val_score(random_forest,
X_train, y_train, cv=8).mean())
```

Decision tree and random forest accuracy

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics
import accuracy_score, classification_report, confusion_matrix
# Make predictions on the test set
pred_dtree = decision_tree.predict(X_test)
pred_rforest = random_forest.predict(X_test)
# Calculate accuracy as percentage
accuracy_dtree = accuracy_score(y_test, pred_dtree) * 100
accuracy_rforest = accuracy_score(y_test, pred_rforest) * 100
# Print accuracies
print(f'Decision Tree Accuracy: {accuracy_dtree:.2f}%')
print(f'Random Forest Accuracy: {accuracy_rforest:.2f}%')
```

```
# Print classification reports
print('Decision Tree Classification Report:\n', classification_report(y_test, pred_dtree))
print('Random Forest Classification Report:\n', classification_report(y_test, pred_rforest))
# Compute confusion matrices
cm_dtree = confusion_matrix(y_test, pred_dtree)
cm_rforest = confusion_matrix(y_test, pred_rforest)
# Plot confusion matrices
fig, axes = plt.subplots(1, 2, figsize=(10, 5))
sns.heatmap(cm_dtree, annot=True, fmt='d', cmap='Blues', ax=axes[0])
axes[0].set_title('Decision Tree Confusion Matrix')
axes[0].set_xlabel('Predicted Labels')
axes[0].set_ylabel('True Labels')
sns.heatmap(cm_rforest, annot=True, fmt='d', cmap='Greens', ax=axes[1])
axes[1].set_title('Random Forest Confusion Matrix')
axes[1].set_xlabel('Predicted Labels')
axes[1].set_ylabel('True Labels')
plt.tight_layout()
plt.show()
```

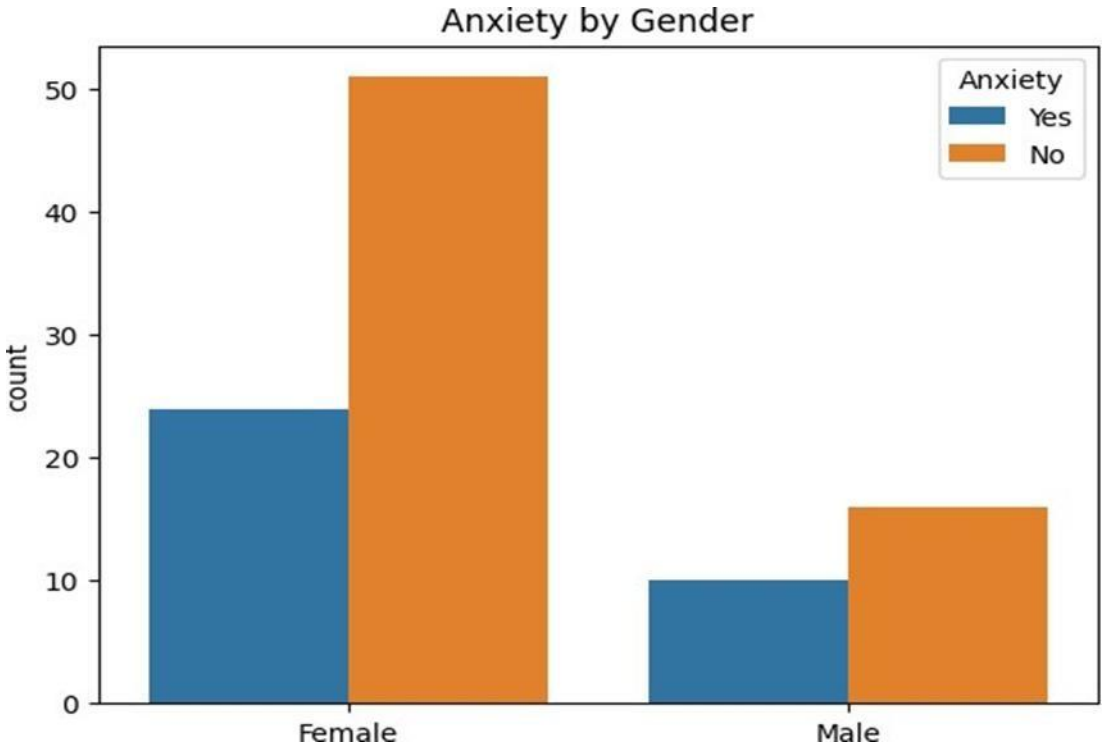
Comparison of algoritms

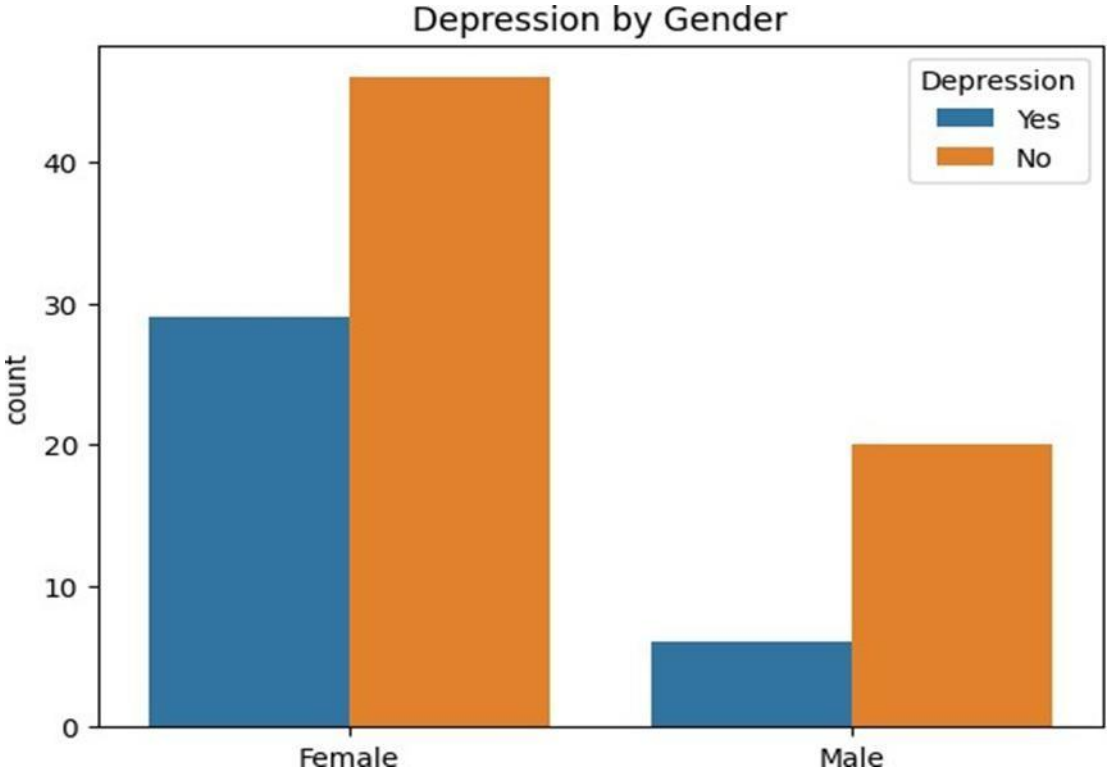
```
import matplotlib.pyplot as plt
# Define model names and accuracies
models = ['Decision Tree', 'Random Forest']
accuracies = [accuracy_dtree, accuracy_rforest]
# Plot the bar chart
plt.figure(figsize=(5, 5))
bars = plt.bar(models, accuracies, color=['skyblue', 'orange'])
```

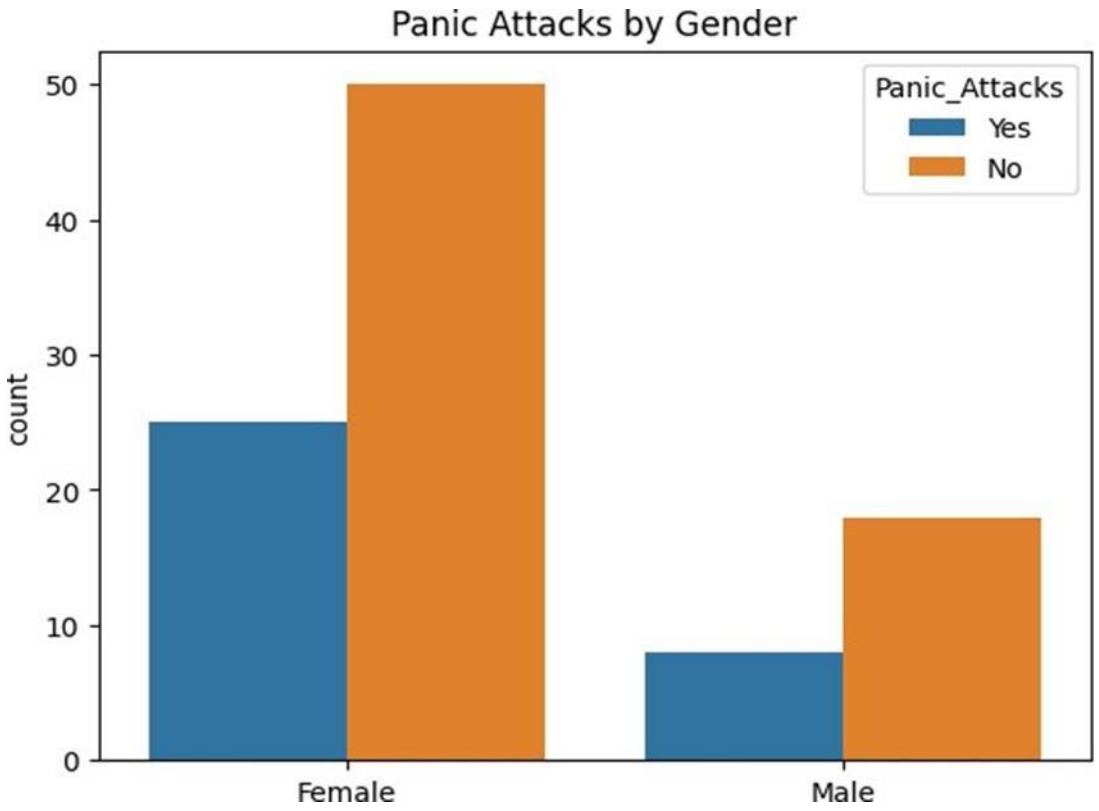
```
# Annotate the accuracy percentages on top of each bar
for bar, accuracy in zip(bars, accuracies): plt.text(bar.get_x() + bar.get_width() /
2,bar.get_height() - 5, f'{accuracy:.2f}%', ha='center', va='bottom', color='black',
fontsize=12)
# Add titles and labels
plt.title('Comparison of Model Accuracies', fontsize=16) plt.ylabel('Accuracy
(%)',
fontsize=12) plt.xlabel('Models', fontsize=12) plt.ylim(0, 100) # Set y-axis
range to 0-100
for clarity
# Highlight which model performs better better_model =
models[accuracies.index(max(accuracies))]
plt.text(0.5, 90, f'{better_model} has better accuracy!', ha='center', color='green',
fontsize=14,
bbox=dict(facecolor='white', edgecolor='green')) # Show the plot
plt.tight_layout()
```

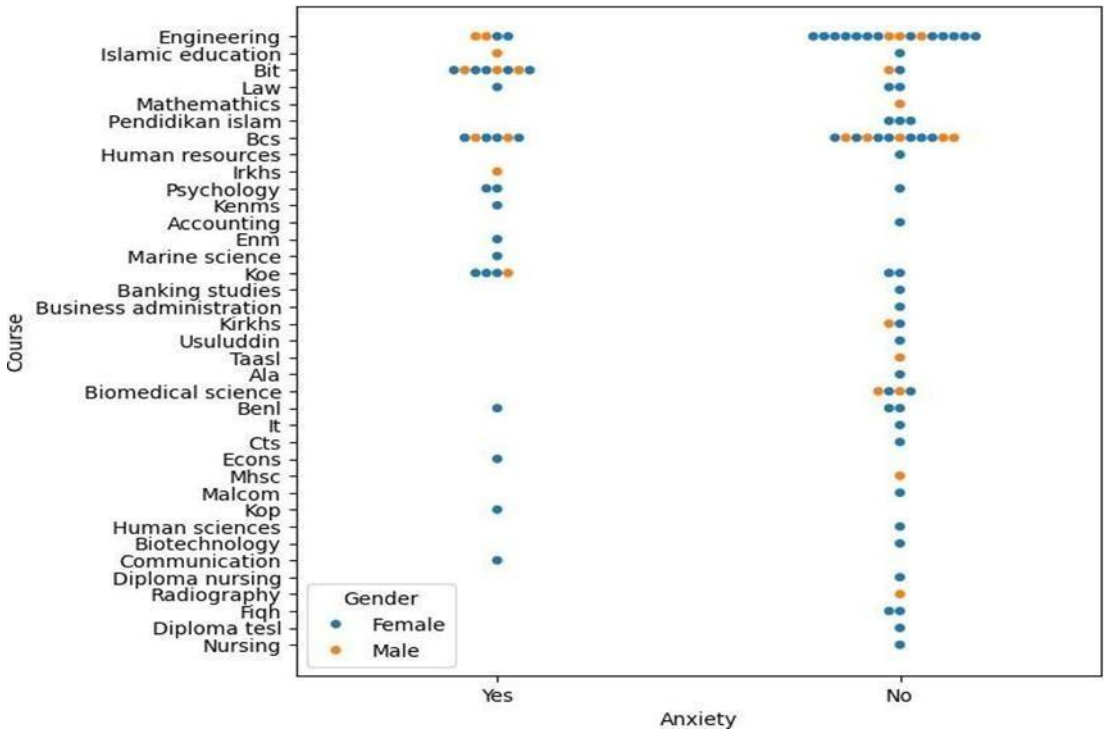
APPENDIX – B

SCREENSHOTS

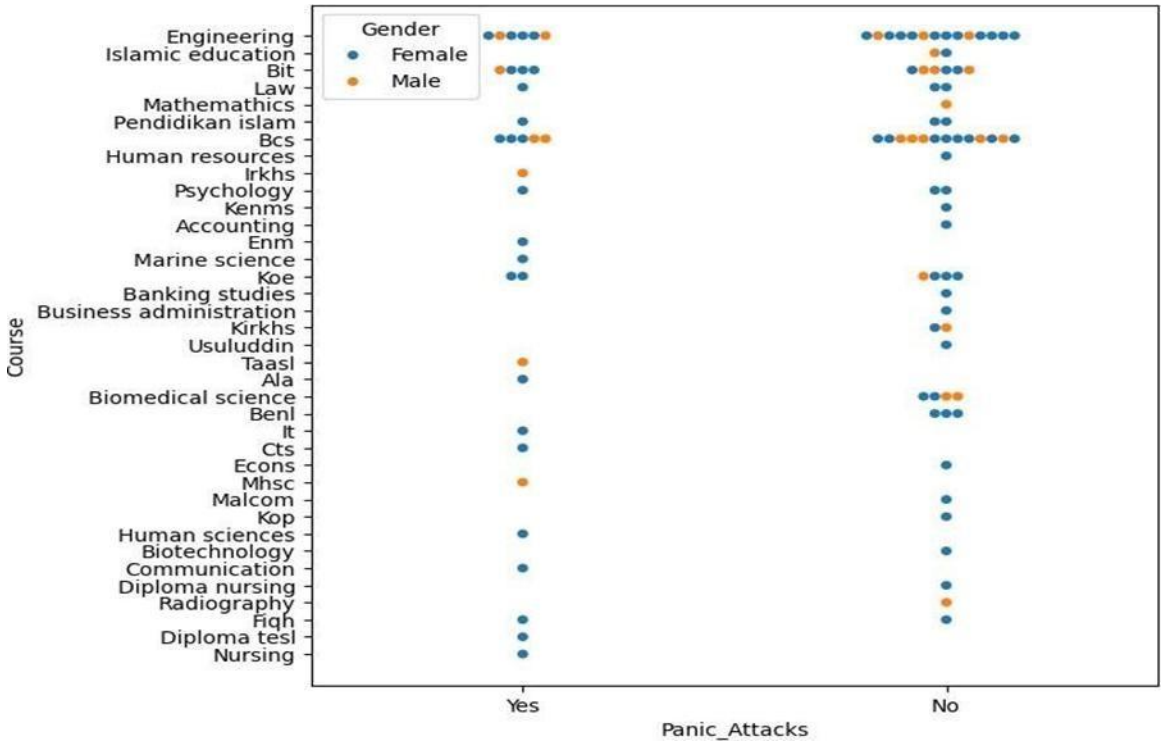





Panic Attacks by Gender
Panic_Attacks
Yes
No
count
50
40
30
20
10
0
Female
Male



Depression by student's course



RESULT ANALYSIS

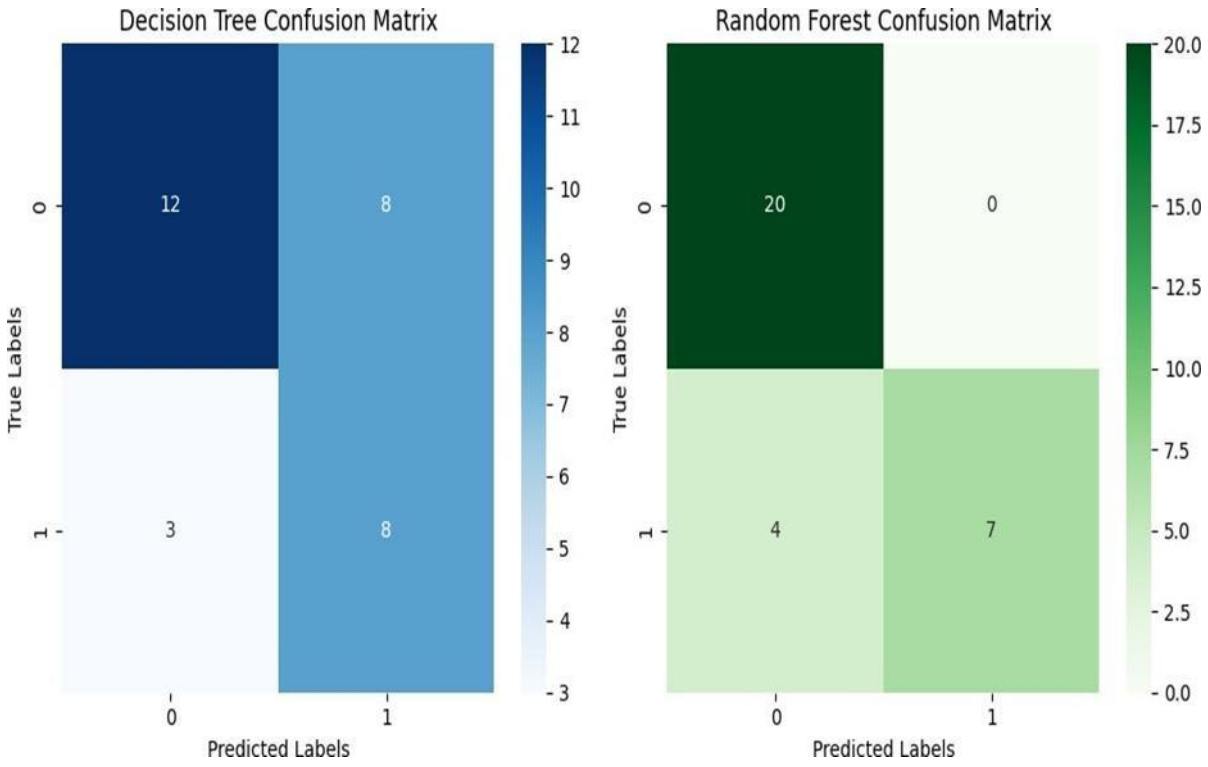
Decision Tree Accuracy: 64.52%
Random Forest Accuracy: 87.10%

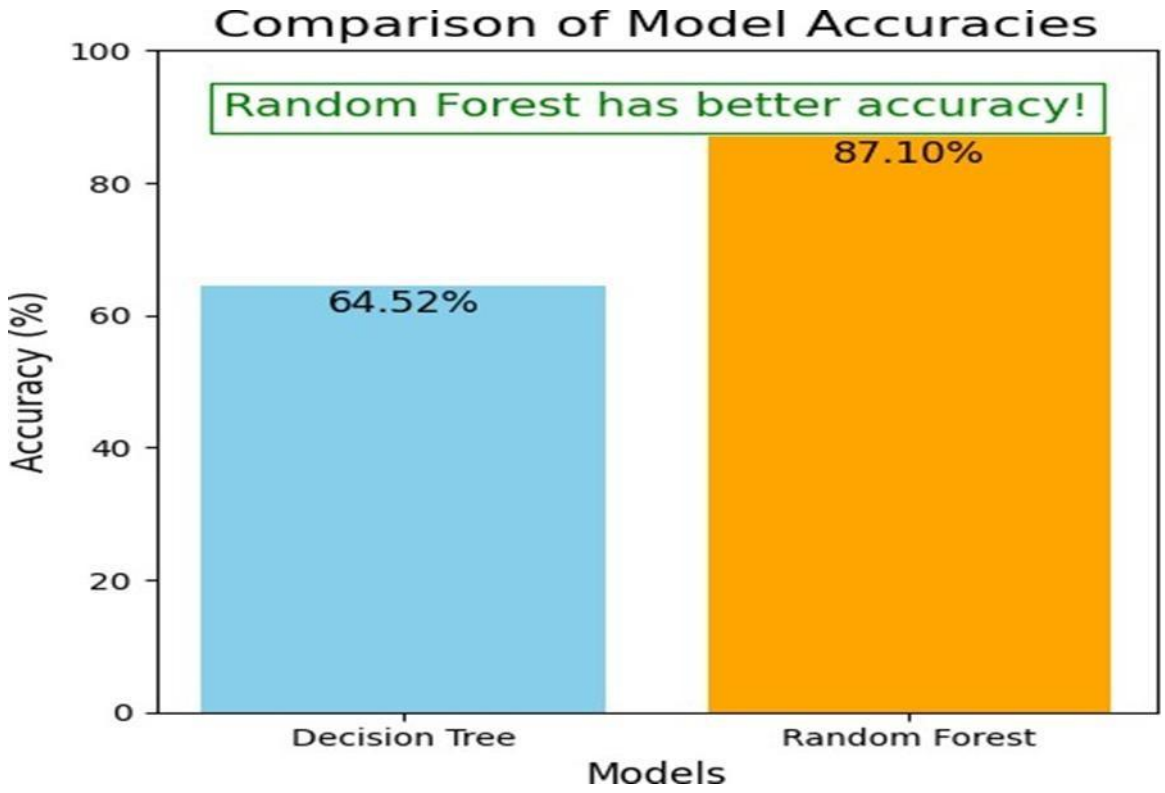
Decision Tree Classification Report:

	precision	recall	f1-score	support
0	0.80	0.60	0.69	20
1	0.50	0.73	0.59	11
accuracy			0.65	31
macro avg	0.65	0.66	0.64	31
weighted avg	0.69	0.65	0.65	31

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.83	1.00	0.91	20
1	1.00	0.64	0.78	11
accuracy			0.87	31
macro avg	0.92	0.82	0.84	31
weighted a	0.89	0.87	0.86	31


Decision Tree Confusion Matrix
True Labels
Predicted Labels
12
8
3
8
0
1
Random Forest Confusion Matrix
True Labels
Predicted Labels
20
0
4
7


Comparison of Model Accuracies
Random Forest has better accuracy!
87.10%
64.52%
Accuracy (%)
Decision Tree
Random Forest
Models

REFERENCES

[1].A.Singh, K. Singh, A. Kumar, A. Shrivastava, S. Kumar, “Machine Learning Algorithms for Detecting Mental Stress in College Students”, arXiv Journal, 2024.

[2].Chowdhury M.R., Xuan W., Sen S., Zhao Y., Ding Y., “Predicting and Understanding College Student Mental Health with Interpretable Machine Learning”, arXiv Journal, 2025.

[3].Fadhluddin Sahlan, Faris Hamidi ,Muhammad Zulhafizal ,Misrat Muhammad ,Haziq AdliSharyar ,WaniYonis., “Prediction of Mental Health Among University Students”, ResearchGate Journel,2021.

[4].Fanglin Xie, Chuhui Geng, Qi Jiang, “Student Mental Health Evaluation System Based on Decision Tree Algorithm”, SpringerLink Conference Paper, 2022.

[5].Jakub Tomasik, Sung yeon sarah han, Jason D.Cooper, “A machine learning algorithm to differentiate bipolar disorder from major depressive disorder using an online mental health questionnaire and blood biomarker data”,IEEE International conference, 2021.

[6].Jetli Chung and Jason Teo,“ Mental Health Prediction Using Machine Learning: Taxonomy, Applications, and Challenges”, ResearchGate Journel,2022.

[7].Konda Vaishnavi , U Nikhitha Kamath , B Ashwath Rao and NV Subba Reddy,“Predicting mental health illness using machine learning”, IEEE International conference,2021.

[8].Ms.Sumathi and Dr.B.Poorna,“Prediction of mental health problems among children using machine learning techniques”, IEEE International conference,2016.

[9].M. Razavi, A. McDonald, R. Mehta, F. Sasangohar, “Evaluating Mental Stress Among College Students Using Heart Rate and Hand Acceleration Data Collected from Wearable Sensors”, arXiv Journal, 2023.

[10].Prathamesh Muzumdar, Ganga Prasad Basyal, Piyush Vyas, “An Empirical Comparison of Machine Learning Models for Student’s Mental Health Illness Assessment”, arXiv Journal, 2022.

[11].Ravinder Ahuja, Alisha Banga, “Mental Stress Detection in University Students using Machine Learning Algorithms”, Procedia Computer Science, 2019.

[12].Rohizah abd rahman, Khairuddin omar, “Application of machine learning methods in mental health detedtion”, IEEE International conference,2020

[13].Sandip Roy , P.S.Aithal , Rajesh Bose,“Judging mental health disorders using the decision tree models”,IEEE International conference,2017.

[14].Sofianita Mutalib, Nurul Aiman Zakaria, Nurul Izzati Jamaluddin, “Mental Health Prediction Models Using Machine Learning in Higher Education Institution”, Turkish Journal of Computer and Mathematics Education, 2021.

[15]. Shaoxiong Ji, Tianlin Zhang, Luna Ansari, Jie Fu, Prayag Tiwari, Erik Cambria, “MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare”, arXiv Journal, 2021.

[16].Satvik Gurjar, Chetna Patil, Ritesh Suryawanshi, Madhura Adadande, Ashwin Khore, Noshir Tarapore., “Mental Health Prediction Using Machine Learning”,International Research Journal of Engineering and Technology (IRJET),2022.

[17].Shumaila Aleem, Noor ul Huda, Rashid Amin, Samina Khalid, Sultan S. Alshamrani, Abdullah Alshehri, “Machine Learning Algorithms for Depression: Diagnosis, Insights, and Research Directions”, Electronics (MDPI), 2022.

[18].Siti Nuarini, Siti Fauziah, N. A. Mayangky, R. Nurfalah, “Comparison Algorithm on Machine Learning for Student Mental Health Data”, Journal of Medical Informatics Technology, 2023.

[19].Salma S. Shahapur, Praveen Chitti, Shahak Patil, Chinmay Abhay Nerurkar, Vijay Shivaram Shivannagol, Vinayak C. Rayanaikar, Vishwajit Sawant, Vadiraj Betageri, “Decoding Minds: Estimation of Stress Level in Students using Machine Learning”, Indian Journal of Science and Technology, 2024.

[20].Xiaohang Xu, Hao Peng, Lichao Sun, Md Zakirul Alam Bhuiyan, Lianzhong Liu, Lifang He, “FedMood: Federated Learning on Mobile Health Data for Mood Detection”, arXiv Journal, 2021.